



Mapeando una entidad #2

Transcripción

[00:00] Hola, continuando con el desarrollo de nuestra entidad, de la entidad producto, ahora vamos a proceder a la parte del id. El id es un elemento existente en las tablas que representa la unicidad de cada una de esas filas, para nosotros indicarle a nuestro proyecto cuál va a ser el id o el identificador de cada una de esas filas, nosotros tenemos otra anotación, es la anotación @Id también de JPA.

[00:28] Importamos la clase de JPA, y es identificador, la responsabilidad de generar ese identificador va a ser siempre de la base de datos. Entonces, para nosotros indicarle que la responsabilidad no pertenece al usuario, sino a la base de datos, tenemos otra anotación que sería la anotación @GeneratedValue y tenemos que pasarle un parámetro, el parámetro de la estrategia de generación que se escribe de esta forma.

[01:00] Strategy=GenerationType. Entonces el tipo de generación para esta base de datos sería la identity. Eso va a depender de la base de datos y eso lo pueden encontrar dentro de la documentación de cada una de las bases de datos. El caso particular de h2 es identity, generalmente se usa identity.

[01:23] Pero puede ser que eso varía para sequency o podrían dejarlo como automático. Nosotros en este caso vamos a usar identity y lo vamos a importar de JPA. Entonces, ya con eso tenemos nuestro id configura para representar la llave primaria en nuestra tabla.

[01:46] Entonces como habíamos mencionado el mapeamiento en Hibernate se realiza de forma automática. Él considera que el nombre de la clase tiene que ser igual al nombre de la tabla. Pero si no es así, nosotros tenemos que usar una anotación para indicar que hay una diferencia en esa tabla.

[02:05] Lo mismo pasa con los atributos. Si nosotros tenemos un atributo que sea diferente tenemos que utilizar la anotación `@Column` pasando el parámetro `name`, igual y pasaríamos el nombre correcto de la tabla. Digamos que sean nombres, pasaríamos nombres. Siempre tienen que ser el mismo valor para poder realizar el mapeamiento. Si no, generaría un error.

[02:29] En este caso podrían dejar con el mismo nombre, pero sería redundante y como es el mismo nombre nosotros vamos a eliminar esta anotación, solamente para que lo sepan. Lo último que está faltando por ahora, es generar los getters y setters para poder trabajar con los valores de cada uno de esos atributos.

[02:52] Entonces, con clic derecho vamos a hacer clic derecho en el panel, vamos a ir a source y vamos a ir donde dice generar getters y setters. Vamos a seleccionarlos todos y los generamos. Ya con eso tenemos nuestra entidad previamente configurada. Vamos a comenzar a trabajar con ella.

[03:14] Solamente nos está faltando un último mapeamiento, vamos a guardar esto acá. En nuestra clase de persistence, nosotros tenemos que realizar otro mapeamiento para Hibernate. En este caso, no tendríamos que hacerlo, pero si ustedes están utilizando otro framework, probablemente tengas que realizar ese mapeamiento.

[03:36] Y el mapeamiento para indicar cuáles son las entidades que van a ser mapeadas en la base de datos es la anotación `@Class`. Entonces, con esa anotación, ustedes van a indicar dónde se encuentra, dónde está la ruta de las entidades que van a ser mapeadas. En este caso sería `com.latam.alura.tienda` y la entidad que va a ser mapeada sería `producto`.

[04:08] Entonces, si ustedes mapean una de las de las entidades tendrían que mapear todas las entidades, no podrían mapear una y otra no. Hibernate, por default, si ustedes no colocan esta etiqueta, él mapea todas las entidades que tengan la anotación @Entity.

[04:30] Si la clase no tiene esa anotación, ella va a ser desconsiderada a la hora de realizar este mapeamento. Así que vamos a guardar esto. Eso sería todo en la parte de creación de entidad. Más adelante nosotros vamos a ir agregando más propiedades, vamos a agregar más atributos y vamos a explicar cómo eso se comporta dentro de nuestro proyecto.